TITLE OF THE INVENTION

Copy Protection of Portable Game Software

RELATED APPLICATIONS

[0001] This application is a continuation-in-part of copending U.S. patent application Serial Number 10/427,793 filed April 30, 2003 and titled "Secure Distribution of Portable Game Software". This application in its entirety is incorporated by reference herein.

BACKGROUND OF THE INVENTION

Field of the invention

[0002] This invention relates generally to electronic game systems and more particularly to game software distributed in encrypted form for copy protection.

Description of the prior art

[0003] Portable game systems that generate player controlled objects in simulated worlds for display on an LCD screen are well known and are described in US patent 6,369,827. It is also well known to store game program instructions and graphics data in digital memory cartridges that plug into such portable game systems. Even if such digital memory cartridges include a trademark and copyright notice as described in US patent 5,184,830, software pirates disregard such notices. Game software in executable form is easily copied and is often sold by software pirates in counterfeit cartridges and disks and is distributed freely on the Internet. It is also known to protect programs by securely storing them in a digital memory in the same processor chip that executes the program instructions as described in US patent 6,339,815. It is also known to include microprocessors in portable

game cartridges as described in US patent applications 2002/0028710 and 2003/0050116. Crypto microprocessors that execute encrypted programs using bus encryption are also disclosed in my US patent 4,278,837.

[0004] Software for game systems has been distributed on laser-readable optical disks for use in game systems. Game software is typically pressed into optical disks during disk fabrication and may be encrypted for copy protection. More than a hundred patents have been issued for optical disks, encryption, and related technologies, such as US patents 6,081,785 and 6,175,629.

[0005] Piracy of portable game software (programs and data) is similar to piracy of music software. When digitized music is read from a data storage medium or decrypted so that it can be converted to analog sounds that can be heard, the digitized music is easy for pirates to copy. But there is one major difference between music and game software. Game programs do not have to be heard or seen by their users and hence game programs do not have to be executed in easily accessible portable game system processors.

[0006] In the present invention, encrypted game programs can be distributed in cartridges and decrypted, stored, and executed in integrated crypto processors in game systems to generate game data, without the game programs being accessible outside of the crypto processors.

SUMMARY OF THE INVENTION

[0007] It is a primary objective of the present invention to provide copy protection for game software that is used in portable game systems. It is another objective of this invention that this protection be provided at low cost and not require a vendor provided game-activation service.

[0008] The preferred embodiment of this invention is an electronic game system for distributing game software in encrypted form in a cartridge together with a crypto processor that contains the decryption key for decrypting the encrypted software. The game system requires a second crypto processor that decrypts the software for execution.

ADVANTAGES

[0009] By distributing game software in encrypted form, proprietary game software can be delivered securely to users without risk that the software will be illegally copied.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a general block diagram of one embodiment that includes a disk cartridge connected to a portable game system.

Fig. 2 is a block diagram of one embodiment that includes a disk cartridge with a first crypro processor connected to a second crypto processor in a portable game system.

Fig. 3 is a block diagram of another embodiment that includes additional encryption.

Fig. 4 is a perspective view of a portable game system with a disk cartridge and crypto processor.

Fig. 4a is a block diagram of a portable game system with an internal crypto processor and an external cartridge.

Fig. 5 is a block diagram of crypto functions used in a disk pressing plant.

Fig. 6 is a detailed block diagram of one embodiment of crypto processor 52.

Fig. 7 is a block diagram of another embodiment that reads an encrypted key from an optical disk.

Fig. 8 is a block diagram of another embodiment that reads an encrypted key from a bar code burned into an optical disk.

Fig. 9 is a block diagram of another embodiment that includes a ROM cartridge with a crypto processor.

Fig. 10 is a perspective view of two human game players operating portable game systems having LCD devices that display multiple articulated body parts of player controlled characters.

Fig. 11 is a perspective view of a portable game system with a crypto cartridge and displaying a 3D image of a player character and a non-player character.

Fig. 12 is a perspective view of a cartridge circuit board having crypto processor 303 and ROM 97 attached to the circuit board.

Fig. 13 is an example of a memory map illustrating software stored in ROM 91 in crypto processor 52.

Fig. 13a is an example of a memory map illustrating software stored in ROM 313 in crypto processor 303.

DETAILED DESCRIPTION OF EXEMPLARY EMBODIMENT

[0010] Fig. 1 illustrates a preferred embodiment of a game system that comprises crypto memory cartridge 16 connected to portable game system 47. The objective is to securely execute and process in portable game system 47 the encrypted software (programs and data) read from tracks 82 of disk 43 housed in cartridge 16. The software on disk 43 (or in semiconductor memory 97 in Fig. 9) is block encrypted using a symmetric digital key K1. Disk 43 is accompanied in each cartridge 16 by a crypto processor 303 (detailed in Fig. 2) that contains key K1 that enables decryption of the encrypted software in crypto processor 52 which is soldered into the circuit board in portable game system 47. Decryption key K1 is not permanently stored in crypto processor 52 or in portable game system 47 and typically would be changed for each game title. When key K1 is changed for encrypting the software on disk 43, key K1 is also changed in crypto processor 303. To prevent accidental separation of disk 43 and corresponding processor 303, both components are housed together in the same cartridge 16.

[0011] Because key K1 is stored and distributed in crypto processor 303 but is used for decryption in crypto processor 52, crypto processor 303 encrypts key K1 and transmits the encrypted key through connectors 247 and 279 and through data bus 93 to crypto processor 52. Crypto processor 52 (detailed in Figures 2, 3, and 6) decrypts the encrypted software read from disk 43 and stores the decrypted software in volatile memory 90 (see Fig. 6) which is on the same chip as processor core 134 in crypto processor 52 which executes the decrypted programs from memory 90. Neither crypto processor 303 nor 52 reveal keys or decrypted program instructions outside of the processor chips.

[0012] Although pirates can make unlimited copies of the encrypted software and perhaps distribute them on the Internet, the encrypted software cannot be decrypted and executed without the two crypto processors 303 and 52 which cannot be copied without knowledge of the inaccessible secret keys.

[0013] Although some of the software for each game may be unencrypted on disk 43 and be loaded into RAM 53 and be executed and processed by conventional processors 50 and 301, this unencrypted software is useless without the decrypted software that is processed by crypto processor 52 to interact with unencrypted software processed by processor 50.

[0014] Decrypted programs executed in crypto processor 52 may generate partially processed game data, such as locations and directions of player controlled objects and points of view. This partially processed game data is further processed by conventional processor 50 and image coprocessor 301 to generate rapidly changing pixel display data in VRAM 302 for display on LCD screen 22.

[0015] Preferably, a small fraction of the programs read from disk 43 would be encrypted and executed by processor 52 while most of the programs from disk 43 would be loaded into RAM 53 and executed by processor 50. Programs that are easy for pirates to reverse engineer need not be encrypted. Programs that are difficult to reverse engineer and do not require rapid access to image co-processor 301 would be suitable for encryption and for execution in crypto processor 52.

[0016] Portable game system 47 may remain an open system for non-crypto cartridges 16 which do not require processor 303 and which store conventional software processed by conventional processors 50 and 301. Prior-art cartridges containing unencrypted programs would not use crypto processor 52.

[0017] Conventional processor 50 is shown in Fig. 1 receiving control data from manual control members such as joystick 20, cross-switch 15, and button-switch 14, but some of the control data should be routed by processor 50 through data bus 93 to crypto processor 52.

[0018] Fig. 2 illustrates an embodiment in greater detail. For clarity, only the encrypted software will be discussed. Encrypted game programs and data are read from tracks 82 on optically readable disk 43 into RAM buffer 97. Accompanying disk 43 is crypto processor 303 containing a non-volatile data memory storing a block of data that includes symmetric key K1 (reference 100), game title identification number 114, and serial number 101.

[0019] Each crypto processor chip 303 in this embodiment has a different serial number that is loaded into the crypto processor chip 303 during assembly of cartridge 16 (see Fig. 5) so that each block of encrypted data on line 61 will be encrypted differently. This prevents pirates from using cryptanalysis that depends on constant unencrypted data such as key K1 and game id for a given game title.

[0020] Block encryption process 147 encrypts the data block (K1, game id, serial) to produce a block of encrypted data on line 61 for transmission to crypto processor 52. The key used for this transmission should not be the same key for every transmission, because this would provide constant encrypted key data that could be distributed on the Internet.

[0021] Instead, a different session key 304 is used every time data is transmitted on line 61 to processor 52. This session key 304 is generated by random number generator 311 in crypto processor 52. To prevent a pirate from supplying an unauthorized encrypted K1 key on line 61 or supplying a bogus session key, the session key 304 is encrypted by block encryption process 306 under control of a symmetrical chip key 131 (key K3).

[0022] Crypto processor 303 has the same chip key K3 so that only genuine processor chips 303 can decrypt the encrypted session key in block decryption process 307. Since random number generator 311 in processor 52 generates a different session key for each transmission to processor 303, the encrypted session key is also different every time process 306 transmits an encrypted session key to processor 303.

[0023] This deprives pirates of a constant session key they need for cryptanalysis. It also prevents pirates from bypassing processor 303 by sending unauthorized data directly into processor 52 on lines 61 and 71. Lines 61 and 71 are part of data bus 93 in Fig. 1 and may be multiplexed.

[0024] To insure that session key 304 is truly random and not pseudo random, a thermal noise source 310 can generate seed for random number generator 311 as described in US patent 4,694,412.

[0025] When the encrypted session key is decrypted by process 307 under control of chip key K3, the resulting plain session key 304 in processor 303 controls block encryption process 147 of key K1, game id, and serial for transmission on line 61 to processor 52.

[0026] When the encrypted block (key K1, game id, serial) is received on line 61 by crypto processor 52, it is block decrypted using the same random session key generated microseconds earlier by random number generator 311.

[0027] Decrypted key K1 is then used by block decryption process 111 to decrypt blocks of encrypted programs and data from RAM 97 to produce decrypted blocks of programs and data that are stored into RAM 90.

[0028] One of the first blocks to be decrypted by process 111 may include the game title identification number in RAM 90 which is then compared to the decrypted game title identification number 114 by verification process 136. If the two game id's do not match, an error message may be displayed on LCD 22.

[0029] Decrypted serial number 101 in crypto processor 52 should not be revealed outside of processor 52 because that would provide pirates with known plaintext to encryption process 147. However, a block encrypted serial number (not shown in Fig. 2) may be displayed on LCD 22.

[0030] Fig. 3 illustrates another embodiment in which the key data block (key K1, game id, and serial) is stored in crypto processor 303 in encrypted form in non-volatile memory 94. Encrypted key K1 is preferably doubly encrypted by block encryption process 147 in processor 303 under control of session key 304. The encrypted data block in memory 94 is previously block encrypted under control of key K2 that is not stored in processor chip 303. This is to deter insider theft of key K1 during manufacturing. Crypto processor 52 doubly decrypts key K1, game id, and serial using session key 304 and chip key 98 (K2).

[0031] Workers in plants that load key data into crypto processor 303 would not know the values of key K2. Workers in plants that load key data into crypto processor 52 would not know the values of key K1. This separation of functions is important for internal security.

[0032] Fig. 4 is a perspective view of portable game system 47 with disk cartridge 16 containing optical disk 43 and crypto processor 303. Serial port 40 is for a link cable to connect portable game system 47 to a console video game system so that portable game system 47 may be used in place of a controller or as an auxiliary display.

[0033] Fig. 4a is a simplified block diagram illustrating portable game system 47 with an internal crypto security processor 52, external memory cartridge 16, and serial port 40.

[0034] Fig. 5 is a block diagram illustrating disk fabrication processes used in a disk pressing plant for writing data onto disk 43 and into crypto processor 303. Random number generator 55 generates a pseudo random symmetrical encryption/decryption key 100 (key K1). Key 100 controls block encryption process 133 which encrypts plain game programs and data 104 to produce encrypted game programs/data 97 which are molded as tracks 82 into disk 43 by disk molding/pressing machine 149.

[0035] Key 100 (key K1), game title identifier 114, serial number 101, and other optional encrypted keys are block encrypted by encryption process 129 under control of key 98 (key K2) selected from key table 110 to produce encrypted block 94 which is stored into crypto processor 303 in the Fig. 3 embodiment. A randomly generated key selection number 113 specifies which key 98 is selected from table 110.

[0036] Adder 64 generates a disk identification serial number 101 which is different for each disk. Key 98 (key K2) is also stored into crypto processor 52.

[0037] Alternatively, the key block may be molded as track 148 into disk 43 (see Fig. 7) by disk molding/pressing machine 149 at the same time tracks 82 are pressed.

[0038] In addition, key selection number 113 (see Fig. 6) may be stored into crypto processor 303 or disk 43.

[0039] Fig. 6 is a detailed block diagram of one embodiment of crypto processor 52. To deter pirates from providing bogus encrypted blocks on line 61, a random bit string is generated by crypto processor 52 and sent to crypto processor 303 which processor 303 immediately alters and returns to processor 52 on line 61. Response timer 314 in processor 52 measures the number of clock cycles between sending the bit stream and receiving the correct response in processor 52. Responses delayed less than m clock cycles or more than n clock cycles are rejected as bogus.

[0040] When decrypted programs and data are stored into RAM 90, processor core 134 executes decrypted program instructions from RAM 134 in addition to instructions stored in boot ROM 91.

[0041] Crypto processor 52 typically executes decrypted programs from memory 90 while cartridge 16 is inserted into portable game system 47. Alternatively, cartridge 16 may be removed from connector 279 and the decrypted game programs in memory 90 may be executed by processor 52 as long as portable game system 47 is electrically powered.

[0042] Communication between crypto processor 52 and conventional processor 50 is through SRAM 239 that is bus multiplexed as described in detail in my copending US patent application serial number 10/427,793 filed April 30, 2003.

[0043] Fig. 7 is a block diagram of another embodiment in which encrypted key K1 is read from track 148 on disk 43 instead of being stored in crypto processor 303. Game identifier 54 is stored into crypto processor 303 for comparison with encrypted game identifier 114 in crypto processor 52 to prevent pirates from using one processor 303 for many different games.

[0044] Fig. 8 is a block diagram of another embodiment in which encrypted key K1, game id, and serial number are read from bar code 80 that is laser burned into optical disk 43 after molding instead of being stored in crypto processor 303. Game identifier 54 is stored into crypto processor 303 for comparison with encrypted game identifier 114 in crypto processor 52 to prevent pirates from using one processor 303 for many different games.

[0045] Fig. 9 is a block diagram of another embodiment in which encrypted programs and data are read from semiconductor memory 97 instead of from optical disk 43. Memory 97 and crypto processor 303 are housed in cartridge 16 (see Fig. 12).

[0046] Fig. 10 is a pictorial view of two human game players 10 and 12 operating portable game systems 44 and 47 having LCD devices that display multiple articulated body parts of player controlled characters, such as arm 59, hand 36, and wrist 37. Player controlled objects are exemplified by pipe 35.

[0047] Fig 11 is a perspective view of portable game system 47 displaying a 3D picture on LCD 22 of two animated characters that have multiple body parts. Movement of the body, arms, and legs of the human-like player character are controlled by manually operated control devices such as cross-switch 15, push button switches 14, and other manually operated devices. One embodiment of cartridge 16 is shown in Fig. 11 inserted into portable game system 47.

[0048] Fig 12 is a perspective view of a cartridge circuit board 299 with crypto processor 303 and ROM 97 attached. ROM 97 can be a separate chip as shown, or it can be included on the crypto processor 303 chip. See Fig. 9 for this embodiment.

[0049] Fig. 13 is an example of a memory map of program instructions and data stored in boot ROM 91 in crypto processor 52 for execution and processing by processor core 134. Some of these instructions may be stored in RAM 90 instead. For example, some of these instructions may copy position data, location data, direction data, and/or texture data from RAM 90 to SRAM 239.

[0050] Fig. 13a is an example of a memory map of program instructions and data stored in boot ROM 313 (not shown) in crypto processor 303.

[0051] In the examples described herein, block encryption and decryption used in processes 99, 111, 142, 147, 306, 307 operate on blocks of at least 64 bits under control of symmetrical keys of at least 64 bits and preferably 128 bits. Such block encryption may use a Feistel-class of block encryption methods such as the Data Encryption Standard (DES), AES, or similar methods, so that changing any one bit of plaintext affects all bits of

ciphertext in an encrypted block, changing any one bit of ciphertext affects all bits of plaintext in a decrypted block, and changing any one bit of an encryption key affects all bits of plaintext in the block, without providing clues that would lead to discovery of the bit values of the secret key through chosen ciphertext attack, chosen encrypted key attack, toggling of bits in ciphertext or encrypted key, differential cryptanalysis, differential fault analysis, and other cryptanalysis techniques. DES is described in detail in Federal Information Processing Standard (FIPS) 46, Nov 23, 1977; and in FIPS PUB 46-3, October 25, 1999.

[0052] DES is considered obsolete because it has been successfully cracked using differential cryptanalysis with massive amounts of plaintext-ciphertext pairs. But in the present invention, there need not be any plaintext-ciphertext pairs. The decrypted programs are stored in RAM 90 and are not revealed outside of crypto processor 52. Likewise much of the data goes no farther than RAM 90 and processor core 134. By encrypting only program instructions and literal data in instructions, but leaving unencrypted the data that is transferred on bus 93 to processor 50 in Fig. 1, there will be no plaintext-ciphertext pairs that a pirate could use in a cryptanalysis attack. Without known plaintext, DES is more than adequate for this application.

[0053] Simplified variations of DES may therefore be used for the present block encryption/decryption processes. The initial permutation IP and the inverse permutation IP^{-1} can be removed because they were apparently designed for use with ASCII text messages. Executable program instructions are not ASCII and are not text messages. The DES key schedule can also be removed because it was designed to limit the key to 56 bits, a limitation that was subsequently relaxed.

[0054] Although encrypted data is accessible on bus 61, encryption of variable session keys prevents access to encrypted-unencrypted pairs. Hence keys K1, K2, and K3 would be very difficult to discover.

[0055] To distinguish encrypted data from non-encrypted data on disk 43, a table can be recorded on disk 43 with tags to indicate which address ranges are reserved for encrypted or non-encrypted data. Tags could also be recorded in header data that precede encrypted data and precede non-encrypted data.

[0056] Symmetric key block encryption uses the same secret key for decryption and for encryption. Typically this key is at least 64 bits and preferably 128 bits or larger. In the preferred embodiment, there is not one master key in processors 303 or 52, because if it were compromised, perhaps by an employee or contractor of the game vendor, the processors would become useless. Instead, in the preferred embodiment, each of crypto processors 303 and 52 includes key table 110 (see Fig. 6) so that secret key K2 and K3 can be changed in mid production of any game title by changing to a different key in the table. If the key bits in table 110 are intermingled with unused random decoy bits, anybody who accesses the bits will not know which bits are key bits without also reading the on-chip ROM or RAM program that access bits that are key among bits that are decoys and reconstruct their sequence.

[0057] Key table 110 in processors 303 and 52 may be stored in an SRAM powered by a battery 130, so that attempts to probe, scan, or peel processor chips 303 or 126 would break a power trace and destroy the keys in table 110. If key table 110 were mask programmed or stored in EEPROM or flash ROM, that would reduce security of the keys, unless the key bits

were rearranged and/or distributed among decoy bits. Keys should not be externally readable or changeable in crypto processors 303 or 52. Key table 110 should be physically protected against probing, chip peeling, scanning electron microscopy, and voltage-contrast imaging. Physical security for chip keys is described in detail in my US patent 4,278,837 for crypto microprocessors that use bus encryption.

[0058] Processor core 134 in the Fig. 6 example, includes an ALU, registers, a stack, instruction decoder, and a program counter to address each executable instruction in sequence in ROM 91 and RAM 90, fetch each instruction, and increment the program counter to address the location of the next instruction.

[0059] Crypto processor 52 in this example, executes decrypted programs stored in RAM 90 that generate intermediate game data that may represent variable characteristics of one or more player controlled objects or characters, and/or non-player objects that move across a background, and/or 2D or 3D views of a simulated world. The game data generated in processor 52 may represent positions, locations, and directions of player controlled game objects such as characters with articulated arms and legs and predefined textures. Even if animation of arms and legs is performed by image coprocessor 301, the spatial coordinates, orientation, and direction of movement of the character may be specified by processor 52 executing the decrypted program instructions in RAM 90.

[0060] The game data generated in crypto processor 52 may also represent positions, locations, and directions of points of view, and may also represent game scores, game status, maps, statistics, object selection, icons, verbal descriptions, instructions, menus, other displayable data, and/or signals to trigger music, voice sounds, and sound effects.

[0061] Data representing background scenery in 2D portable game systems may be unencrypted on disk 43 and loaded into RAM 53 because backgrounds are easily readable or easily reconstructed by pirates. But the program instructions that determine when and what backgrounds are needed and what changes are made to backgrounds (such as a door remaining open) may be executed by crypto processor 52 from RAM 90.

[0062] Image coprocessor 301 in 2D systems may perform scrolling, flipping, blending, scaling, rotating, fading, windowing, and other image processing functions on display data stored in display RAM (VRAM) 302 for display on LCD screen 22. In 3D systems, image coprocessor 301 may perform coordinate transformations of polygons, texture rendering, bump mapping, lighting and shadows, and rasterizing polygon data into displayable pixel data in VRAM 302 for display on LCD 22.

[0063] As used herein, the term "molded" includes injection molded, pressed, stamped, and other disk fabrication methods.

[0064] The term "video" includes composite, non-composite, RGB, monochrome, color, analog, digital, raster scanned, and the like.

[0065] The details of cartridge 16 and crypto processors 303 and 52 are given here only as examples and numerous other designs may be used.

[0066] The term "LCD" (liquid crystal display) has been used herein as an illustrative example of any display apparatus having discrete dot-matrix picture elements.

[0067] The term "program" as used herein may consist of more than one loadable module and typically includes executable instruction data and any data that is typically part of a program module or modules.

[0068] The processes of encryption and decryption specified herein may be performed by software, i.e. program instructions executed in a processor core with data tables such as the S-boxes (substitution boxes) that are used in DES. Or the encryption and decryption may be performed in dedicated crypto hardware in the processor chips, or a combination of hardware and software in the processor chips.

[0069] Although I have described my invention with a degree of particularity in connection with what is presently considered to be the most practical and preferred embodiments, the foregoing description has been made only by way of illustration and example and is not to be interpreted as restrictive or limiting as to the meaning of words in the patent or its claims. It is understood that various modifications, variations, arrangements, and/or equivalents, can be devised without departing from the spirit and scope of the invention which is defined by the claims.

Reference Numbers in Drawings

10	human game player
11	television (TV) set or video monitor
12	human game player
14	control switch
15	manual cross-shaped control switch
16	memory cartridge
20	joystick
22	LCD screen
33	LCD pictures
34	LCD pictures
35	player controlled object
36	simulated hand & arm
37	simulated hand & arm
40	serial link port in portable system
43	optical disk
44	portable game system
45	cable from controller to console
47	portable game system
49	cursor
50	processor in portable system
52	crypto processor
53	RAM in portable system
54	game id
55	program process
56	video screen
59	cursor
61	data bus connecting crypto processor
64	program process
71	data bus connecting crypto processor
76	boot ROM in portable system
80	burst cutting area (BCA) of disk
82	tracks molded into disk
83	optical disk reader
90	SRAM in crypto processor

91 boot ROM in crypto processor
92 address bus
93 data bus
94 encrypted block (key K1, game id, serial)
97 encrypted programs and/or data
98 secret key (K2)
99 process of block decryption (K2)
100 secret key (K1)
101 disk serial number
104 plain program(s) and/or data
110 table of keys
111 process of block decrypting (K1)
113 key selection number
114 game product number
119 LCD driver
124 encrypted key (K1), serial
129 process of block encryption (K2)
130 electric battery
131 secret key (K3)
133 process of block encryption (K1)
134 processor core in crypto chip 52
136 verify game id
142 process of block decryption (K4)
147 process of block encryption (K4)
148 lead-in encrypted key
149 disk molding machine
230 insertion slot for cartridge
239 SRAM shared
247 multiple contact connector
279 multiple contact connector
299 cartridge circuit board
300 direct memory access (DMA)
301 image coprocessor
302 video RAM
303 crypto processor
304 session key (K4)

306 process of block encryption (K3)
307 process of block decryption (K3)
310 thermal noise source
311 generate session key
313 boot ROM in crypto processor 303
314 response timer